

**1 (True or False) and Justify**

[20 bodov]

1. Ak by sme pri MergeSorte delili pole na 4 približne rovnaké časti, ešte stále by mal časovú zložitosť  $\Theta(n \log n)$ .
2. Majme dátovú štruktúru vektor, ktorá už obsahuje  $n$  prvkov. Potom postupnosť operácií „vloží prvok na koniec“ a „odstráni prvok z konca“ má vždy časovú zložitosť  $O(1)$ .
3. Majme dátovú štruktúru vektor, ktorá už obsahuje  $n$  prvkov. Potom postupnosť operácií „odstráni prvok z konca“ a „vloží prvok na koniec“ má vždy časovú zložitosť  $O(1)$ .
4. Ak máme v úlohe o stabilných manželstvách  $m$  mužov a  $z$  žien, tak existuje množina  $\min(m, z)$  disjunktných manželstiev, ktorá je stabilná.
5. Z neusporiadaného poľa veľkosti  $n$  ( $n > 47$ ) sa v čase  $O(n)$  dá vyrobiť binárny vyhľadávací strom s hĺbkou  $\leq n/3$ .
6. Máme dve polia veľkosti  $n$ . V čase  $O(n \log n)$  sa dá zistiť, či obsahujú tú istú množinu prvkov.
7. Máme dve polia veľkosti  $n$ . V čase  $O(n \log n)$  sa dá zistiť, či existuje nejaký prvok, ktorý leží v oboch poliach.
8. Máme usporiadané pole  $n$  celých čísel. V čase  $O(\log n)$  sa dá zistiť, koľko z nich je kladných.

**2 Pohľad zvnútra: halda**

[5 + (6 alebo 10) bodov]

- a) Popíšte a/lebo nakreslite, ako presne by ste do poľa uložili klasickú binárnu haldu (s minimom v koreni).
- b1) Napíšte ako pseudokód alebo kus programu funkciu, ktorá do takto uloženej haldy vloží nový prvok.
- b2) Napíšte ako pseudokód alebo kus programu funkciu, ktorá z takto uloženej haldy vyberie najmenší prvok. (Vyberte si jedno z b1 a b2. Za b2 je viac bodov.)

**3 Pohľad zvnútra: hľadanie maxima a minima**

[7+4+3+6+5 bodov]

Ibrahim má pole  $a$  v ňom  $n$  prvkov, nie nutne navzájom rôznych. Pre jednoduchosť budeme predpokladať, že  $n = 2^k$ .

- a) Dokážte, že na nájdenie maxima je vždy potrebné spraviť aspoň  $n - 1$  porovnaní. Inými slovami, dokážte, že pre každý algoritmus, ktorý spraviť menej ako  $n - 1$  porovnaní, existuje vstup, pre ktorý nedá správnu odpoveď.
- b) Ibrahim potrebuje nájsť aj maximum, aj minimum. Keby hľadal každé zvlášť, potrebuje  $2n - 2$  porovnaní. On ale vymyslel algoritmus, ktorý naraz nájde obe: Rozdelíme pole na dve polovice, samostatne v každej nájdem maximum a minimum a z tých potom vypočítame globálne maximum a minimum. Kostra jeho programu:

```
def minmax(A):
    ak dĺžka A je 2:
        <niečo sa stane>
    inak:
        min1, max1 = minmax(prvá polovica A)
        min2, max2 = minmax(druhá polovica A)
        <niečo iné sa stane>
```

Doplňte optimálne časť „niečo sa stane“.

- c) Doplňte optimálne časť „niečo iné sa stane“.
- d) Ibrahimov algoritmus na poli s  $n$  prvkami spraviť  $an + b$  porovnaní. Nájdite  $a$  a  $b$ . Ak  $a \geq 2$ , niečo robíte zle.
- e) Zuzanka sa opýtala: prečo sme ako základný prípad použili „dĺžka A je 2“ a nie až „dĺžka A je 1“? Odpovedzte.

**4 Pohľad zvonka: lenivý stavbár**

[15 bodov]

Ukrajinec Sergej betónuje terasu, ale sa mu nechce. Spraviť a vyliat za miešačku betónu mu trvá presne hodinu. Pozná presné časy  $t_1, \dots, t_n$  (v sekundách od začiatku zmeny), kedy sa na neho príde pozrieť stavbyvedúci. Napíšte (ako pseudokód alebo kus programu) algoritmus, ktorý vypočíta, koľko najmenej miešačiek betónu musí Sergej spraviť, aby ho stavbyvedúci zakaždým videl, ako pracuje. Dokážte správnosť vášho programu a odhadnite jeho časovú zložitosť.

Príklad: ak príde kontrola o 9:47, 10:12 a 13:50, tak stačia dve miešačky: napr. od 9:30 do 10:30 a od 13:47 do 14:47.

**5 Pohľad zvonka: jednozubé pole**

[10 alebo 5 bodov]

Pole  $A[0..n - 1]$  voláme *jednozubé*, ak existuje nanajvýš jedno  $i$  také, že  $A[i] > A[i + 1]$ . (Každé jednozubé pole je teda takmer usporiadané vzostupne.) Eleonóra potrebuje často usporadúvať veľké polia. Navyše však vie, že veľa spomedzi jej polí je jednozubých. Preto sa jej nepáči MergeSort, ktorý má na každom vstupe veľkosti  $n$  časovú zložitosť  $\Theta(n \log n)$ . Rozhodnite a dokážte, či existuje triediaci algoritmus s nasledujúcimi dvomi vlastnosťami:

- Na každom možnom poli veľkosti  $n$  spraviť  $O(n \log n)$  krokov.
- Ak je ale vstupné pole jednozubé, bude časová zložitosť nášho algoritmu len  $O(n)$ .

(Ľahšia verzia za 5 bodov: zmeňte si v predchádzajúcej vete slovo „jednozubé“ na slovo „usporiadané“.)

**6 Pohľad zvonka: strojový čas**

[15 bodov]

Kleofáš má na internete pripojenú mašinu. Rozhodol sa zarábať tým, že ju bude prenajímať na rôzne účely, podobne ako to vo veľkom robí Amazon so svojim Elastic Compute Cloud.

Teraz mu prichádzajú rôzne požiadavky. Každú požiadavku tvoria dva timestampy  $t_1 \leq t_2$ , hovoriace, že zákazník má záujem používať mašinu v uzavretom intervale  $[t_1, t_2]$ . Ak Kleofáš ešte vie požiadavku prijať (t.j. požadovaný interval má prázdny prienik so všetkými skôr prijatými požiadavkami), prijme ju, inak ju odmietne.

Ako pseudokód alebo kus programu napíšte algoritmus, ktorý bude čo najefektívnejšie spracúvať požiadavky. Zdôvodnite jeho správnosť a odhadnite časovú zložitosť v závislosti od celkového počtu požiadaviek  $n$ .

Príklad: pre požiadavky  $[3, 7]$ ,  $[12, 15]$ ,  $[2, 10]$ ,  $[4, 13]$ ,  $[9, 10]$  by prijal 1. a 2., odmietol 3. a 4., a prijal 5. požiadavku.